All Reviewers: We thank all the reviewers for your valuable time and insightful reviews. We also thank your kind appreciation for our ideas and experimental setup to show inductive extension and speed up in the original quantum embedding proposal. We have tried our best to clarify your questions. [A cryptic form of reviewer's question/comment precedes our response.]

Reviewer #2:
[Accuracy comparison with the original method of Garg et al.] We believe we had given the accuracy numbers for the original method of Garg et al. in Table 4 (first row) of the main paper. Here are the performance numbers for the original method (Garg et. al.): **0.383 (accuracy), 0.420 (Macro F1), 0.347 (Micro F1)**. Our performance number on the same task are as follows: **0.631(accuracy), 0.764 (Macro F1), 0.724 (Micro F1)**. Also, we would like to add, in this paper, we are working with fine grained entity classification task which was not considered in the original paper.
[Rotational invariance of the embedding is almost necessitated.] We agree, prima facie, the observation regarding rotational invariance property may not be very surprising. However, we explicitly called this out because of two reasons: (i) it plays a crucial role in the formulation as well as the solution of the subproblem 3 (line # 180-184 in the main paper), (ii) moreover; we felt this property might not strike to a reader's mind in an obvious manner while glancing through the nonlinear and non-convex formulation of the QE problem.
[The discussion at the end of Section 3, mostly contributes to understanding the role of various optimization terms and hyperparameters rather than the embedding itself.] Yes, you are right. However, we added this section, realizing that such a discussion can tremendously help a reader understand and appreciate the formulation as well as the solution of the subproblem 3, which otherwise may sound a bit intricate. We felt such a discussion could offer an intuitive and geometric feel in reader's mind regarding why an axis is chosen for a specific subspace by our algorithm. Also, the reader can convince oneself that the behavior of our proposed algorithm for subproblem 3 is indeed a natural way of generating concept subspaces.
[A gentler introduction to Knowledge Representation and the associated tasks would have been appreciated.] Thanks for pointing it out. We will certainly add a gentle introduction to these topics in the camera-ready version.

Reviewer #3
[Whether the current formulation is the only choice. It would be more convincing if one can show other simple alternatives of IQE is less promising.] This is a very good suggestion. Although we had given a careful thought while formulating the optimization problem (especially, in designing the rank constraints and the orthogonality constraints), we will shed more light on this aspect in the final version of the paper. Just to add, in sections 6.1 and 6.2 of the supplementary material, we have worked out two different simplified versions of the IQuE problem by relaxing either of these two constraints (orthogonality and rank). During our experiments, as stated in line # 126-128 of the main paper, we found that these simpler formulations result in degenerate solutions for the subproblem 3, where multiple concept subspaces get collapsed to zero and thereby resulting in inferior quality embedding.
[The review had to read the original paper to see why inductive reasoning was not implemented in [Garg 19].] We are sorry for the inconvenience. Thanks for pointing it out. In the camera-ready version, we will certainly elaborate on the task implemented (and their non-inductive nature) in the original paper of Garg et al.

Reviewer #5
[While the algorithm is shown to be faster than QE, the comparison does not take into account the time for training the NN.] We want to highlight that NN comes after QE in our pipeline (as shown in Figure 2). The NN part learns the mapping from the input sentence feature vector to the QE, irrespective of which method was used to generate the QE (our method or the original method). Therefore, we felt the meaningful comparison would be to compare only in terms of the time taken to generate QE (by the original method of Garg et al. versus our proposed method). However, for the sake of completeness, we are providing below a table comparing two approaches including the training time of NN as well. In this table, t_{qe} and i_{qe} denote *per iteration time* and *number of iterations*, respectively, taken during the training of any QE method. The quantities t_{nn} and i_{nn} denote *average time per epoch* and *number of epochs* respectively, for the training of NN. The average time t_{nn} is approximately the same for both the methods. There are now two speedup factors - one without including T_{nn} and other with including T_{nn}.

Method	t_{qe}(in sec.)	i_{qe}	$T_{qe} = t_{qe} \times i_{qe}$	t_{nn} (in sec.)	i_{nn}	$T_{nn} = t_{nn} \times i_{nn}$	$T_{qe} + T_{nn}$
Ours	510.6	6	3063.6	$\approx$ 1275	6	7650	10713.6
Garg et. al. 2019	27.8	1000	27800.0	$\approx$ 1275	6	7650	35450.0
Speedup Factor			**9.07**				**3.31**

[The performance on FIGER are second best for F1, even though the method is more generally applicable than the best one for F1.] Yes, you are absolutely right. Even though our method achieved second best F1 on fine-grained entity task, unlike the best F1 method [i.e. *Attentive* (Shimaoka 2016)], our method is task agnostic and is more generally applicable to many downstream tasks. Also, note that our method beats the *Attentive* method in terms of accuracy.